

Importing Packages

First, download and import **Photon2**. Pass your AppID to Photon setup (explained below), and then import the “**RCC_PhotonNecessaryScripts**” in the **Scripts/Photon** folder. You can also import the integration package from the welcome window too. **Tools → BCG → RCC → Welcome Window → Addons**. After importing Photon2 and integration packages to the project, you can test the Photon demo scenes. Three photon demo scenes can be found in the demo scenes folder. Or you can open them via welcome window.

There should be a new scene named “**RCC City Photon2**” in the **Demo Scenes** folder after importing the integration package. Same scene with regular city scene. Only difference is, this scene has **RCC_PhotonDemo** attached to the **RCCCanvasPhoton**. Connects to the photon server, enters lobby, creates or join room, and spawns the vehicle.

How Photon Works

Well, it's quite easy. If you don't know anything about Photon, or even multiplayer based game, here is the events for how it works;

- 1** – First, you have to connect to the server. In this case, our server is **Photon Cloud Server**.
- 2** – If your connection to server succeeded, you will be in lobby with callbacks. Now you can create your room, or join someones room, or join randomly room here.
- 3** – Congrats, now you are in an online multiplayer based room in realtime.
- 4** - Each gameobject with the PhotonView component will be synchronized over the network. **RCC_PhotonNetwork** is syncing all vehicle inputs with PhotonView over the network.

Installing Photon

As soon as you have imported it to your project, it will ask your AppID. You must register and create new app id for your project. Pass it. And now you are ready to develop your realtime based multiplayer levels. It's free. And of course you are limited with 20 CCU.

Photon and RCC

Photon2 has many simple methods in their API. It's extremely easy to understand. Let me explain how the demo scene works;

City demo scene has a **RCCCanvasPhoton** with script named "**RCC_PhotonDemo**". This script handles multiplayer section of the scene. Uses these methods (You can find all methods from Photon2's docs);

Photon.Pun.PhotonNetwork.ConnectUsingSettings ();

We are connecting to the server first. We can listen which connection status we are on in OnGUI() method. Like this;

```
GUILayout.Label("State: " + Photon.Pun.PhotonNetwork.NetworkClientState.ToString());
```

Once connection to the server established, we have to join lobby.

Photon.Pun.PhotonNetwork.JoinLobby ();

As soon as we are connected to lobby, we want to join a random room by;

Photon.Pun.PhotonNetwork.JoinRandomRoom();

If it fails, this means there are no any active other room. We are creating the new room by **Photon.Pun.PhotonNetwork.CreateRoom(null)**; This method needs room name. I didn't use it, because there are no any room lists in the demo.

I take a string that belongs to player here. And set it by

Photon.Pun.PhotonNetwork.NickName = name; Enabling/disabling few UI gameobjects depends on connection state. That's basically, how the demo scene works.

For vehicle sync, each vehicle has **PhotonView.cs** and **RCC_PhotonNetwork.cs**. These scripts are necessary for each network vehicle. **RCC_PhotonNetwork.cs** is observed by the **PhotonView.cs**.

[RCC_PhotonNetwork.cs](#) is synchronizing all control inputs, position, rotation, and rigid velocity smoothly. If vehicle is our vehicle, it will broadcast our data to the server. If vehicle is not our vehicle, it will receive all data from the server.

These vehicles are not instantiated or destroyed with regular **GameObject.Instantiate** or **Destroy**. You have to do it with **PhotonNetwork.Instantiate** or **Destroy**. Unfortunately, it won't work with your prefab. It accepts only strings for your vehicle. That means, it will use **Resources** folder for accessing your vehicles. Your vehicle prefabs must be at **Resources** folder. Therefore, there are two canvases in resources folder. One of them is using [RCC_Demo.cs](#), other one is using [RCC_PhotonDemo.cs](#).

Photon Lobby

[RCC_PhotonManager](#) script handles multiplayer section of this scene. Connects to the photon server, enters lobby, lists current rooms, creates or join rooms. You can find many commented lines in this script. [RCC_PhotonManager](#) and [RCC_PhotonDemo](#) scripts are not same. Manager script handles lobby scene, and demo scene handles actual city demo scene where you can spawn your vehicles.